WL-TM-94-3130

A Non-Iterative Grid Deformation Algorithm for Computational Fluid Dynamics for Aeroelasticity

Richard D. Snyder

Structural Dynamics Branch
Structures Division

October 1994
Interim Report for Period June 1994 – July 1994

Approved for public release; distribution is unlimited

FLIGHT DYNAMICS DIRECTORATE
WRIGHT LABORATORY
AIR FORCE MATERIEL COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433-7562

Notice

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely Government-related procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may be any way related thereto.

This technical report has been reviewed and is approved for publication.

Richard D. Snyder
RICHARD D. SNYDER
Research Aerospace Engineer
Aeroelasticity Section
Structural Dynamics Branch

Terry M. Harris
TERRY M. HARRIS
Technical Manager
Aeroelasticity Section
Structural Dynamics Branch

Joseph W. Moschler, Jr
JOSEPH W. MOSCHLER, MAJOR, USAF
Chief, Structural Dynamics Branch
Structures Division

If your address has changed, if you wish to be removed from our mailing list, or if the addressee is no longer employed by your organization, please notify WL/FIBGE; 2130 Eighth Street Suite 1; Wright-Patterson Air Force Base, OH 45433-7542 to help us maintain a current mailing list.

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
September 1994

3. REPORT TYPE AND DATES COVERED
Interim Report for June 1994 - July 1994

4. TITLE AND SUBTITLE
A Non-Iterative Grid Deformation Algorithm for Computational Fluid Dynamics for Aeroelasticity

5. FUNDING NUMBERS
PE 62201
PR 2401
TA TI
WU 00

6. AUTHOR(S)
Richard D. Snyder

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Structural Dynamics Branch
Structures Division
Flight Dynamics Directorate
Wright Laboratory
Wright-Patterson Air Force Base, OH 45433-7006

8. PERFORMING ORGANIZATION REPORT NUMBER
WL - TM - 94 - 3130

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Flight Dynamics Directorate
Wright Laboratory
Air Force Materiel Command
Wright-Patterson Air Force Base, OH 45433-7562

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
WL-TM-94-3130

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This Technical Memorandum presents an algorithm for deforming a two- or three-dimensional aerodynamic grid given deflections on a physical boundary. This algorithm is to be used in coupling a computational fluid dynamics code with a computational structural mechanics code. The deformation algorithm was tested successfully on both two- and three-dimensional grids. The method sufficiently maintains grid quality for smooth deflections of realistic orders of magnitude. The method is computationally efficient: the time required to deform the grid is small compared to the time required to solve the fluid dynamics.

14. SUBJECT TERMS
Grid Generation
Grid Adaptation
Grid Deformation

15. NUMBER OF PAGES
23

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
Unclassified	Unclassified	Unclassified	UL

Foreword

This Technical Memorandum was prepared by the Aeroelasticity Section of the Structural Dynamics Branch, Structures Division, Flight Dynamics Directorate, Wright Laboratory, Wright-Patterson Air Force Base, Ohio. The study reported herein was conducted under Work Unit 2401TI00, "Structural Technology Integration".

This manuscript was released in September 1994 for publication as a Technical Memorandum covering work performed during June and July 1994.

Summary

This Technical Memorandum presents an algorithm for deforming a two- or three-dimensional aerodynamic grid given deflections on a physical boundary. This algorithm is to be used in coupling a computational fluid dynamics code with a computational structural mechanics code. This algorithm is intended to assist engineers in the Flight Dynamics Directorate with the analysis of aeroelastic problems using state-of-the-art computational methods. The deformation algorithm was tested successfully on both two- and three-dimensional grids. The method sufficiently maintains grid quality for smooth deflections of realistic orders of magnitude. The method is computationally efficient: the time required to deform the grid is small compared to the time required to solve the fluid dynamics.

Table of Contents

List of Figures

Nomenclature

Symbol	Description
r	Distance between interior grid point and surface point
r_0	Smoothing parameter
r_1	Smoothing parameter
$\bar{u}$	Grid point deflection in physical coordinates
$\tilde{u}$	Transformed grid point deflections
U	Magnitude of the maximum deflection in the aerodynamic grid
$\bar{x}$	Grid point location in physical coordinates
$\bar{\xi}$	Grid point location in computational coordinates
ξ_1, ξ_2, ξ_3	Computational coordinates
θ_1, θ_2, θ_3	Rotation angles
Θ_1, Θ_2, Θ_3	Transformation matrices
Subscripts	
a	Set of Interior grid points
b	Set of Boundary grid points
max	Maximum value of a coordinate direction
min	Minimum value of a coordinate direction
n	Arbitrary index
Superscripts	
m	CFD-CSM loop iteration counter

Introduction

The Flight Dynamics Directorate is conducting research towards integrating computational fluid dynamics (CFD) codes with computational structural mechanics (CSM) codes. This will give the Directorate the ability to develop solutions to aeroelastic design and analysis problems using state-of-the-art computational methods. The integration of the CFD and CSM equations is done cyclically. The fluid dynamics equations are solved time-accurately for a single time step. Next, the deflections of the body are computed by the CSM code using surface pressures computed by the CFD code. The new surface deflections are then used to update the aerodynamic grid. With the updated grid, the CFD code is executed for another time step and the process repeats. This paper presents an algorithm for updating the grid point positions of the aerodynamic grid.

An algorithm has been developed for deforming aerodynamic grids that is based on a spring analogy (References 1 and 2). This method, however, is seen to have two limitations. The first is its reliance on grid connectivity. The computer code for a structured grid would be different from that for an unstructured grid and the code for a two-dimensional grid would be different from that for a three-dimensional grid. The second limitation is that the grid point deflections are a function of local grid point spacing, rather than body geometry and global distances. Consequently, the method best maintains orthogonality in regions of high grid point densities, regardless of the proximity to the surface.

A second algorithm has been developed by this author that eliminates the restrictions of the spring analogy method (Reference 3). The computer code can be applied to a wide variety of grid types and body geometries without modification. The algorithm can accurately solve for large deflections. Also, the orthogonality of grid lines at a surface is made a function of the global geometry: the deflection at a given point in the aerodynamic grid is a function of the deflection of the entire aerodynamic surface. However, this global geometry dependence is computationally expensive. For three-dimensional grids, the computational cost is prohibitive. Another algorithm is needed which is both computationally efficient and able to produce a well-deformed grid.

Presented herein is a third grid deformation algorithm that is based on a single first-order differential equation. The smoothing at a point is dependent on the distance from the body. The equation is non-iterative and is solved by marching away from the aerodynamic body. The method is computationally inexpensive, yet maintains the quality of the original undeformed grid. The algorithm is dependent on the grid topology, however. As such, the grid deformation code must be specialized to a particular grid topology.

Grid Deformation Algorithm

The deformation algorithm solves a non-linear relationship between known (independent) boundary deflections and unknown (dependent) interior grid point deflections. The deformation algorithm is a non-iterative method based on an ad-hoc first-order differential equation. The algorithm is applied at each iteration of the CFD-CSM loop.

The algorithm can be divided into five steps:

1. If desired, transform the known grid point deflections to deflections with respect to a body-fixed coordinate system.
2. Determine appropriate marching directions.
3. Solve for the unknown grid point deflections.
4. If necessary, transform the grid point deflections back to the global coordinate system.
5. Update the grid point positions.

Each step is described in the following sub-sections.

Preliminaries

The aerodynamic surfaces and the surrounding grid are defined in terms of physical coordinates $\bar{x}$. To facilitate the solution of the CFD equations, the grid is mapped onto a uniform rectilinear coordinate system $\bar{\xi}$. The engineer constructs this mapping by using an analytic or computational grid generator. Figure 1 shows a sample mapping. A C-H grid about a flat plate is mapped from physical coordinates $\bar{x}$ to computational coordinates $\bar{\xi}$. The flat plate, shaded gray, maps onto a portion of the $\xi_{3_{\min}}$ face of the computational domain. Diagonal hash marks identify the region behind the flat plate.

A grid deformation algorithm must maintain the quality of the original grid. The following conditions should be satisfied for grid quality to be maintained:

a) No negative cell volumes.
b) Grid point clustering is maintained.
c) Orthogonality of grid lines at a physical surface is maintained.
d) Minimal change in cell volumes near a physical surface.

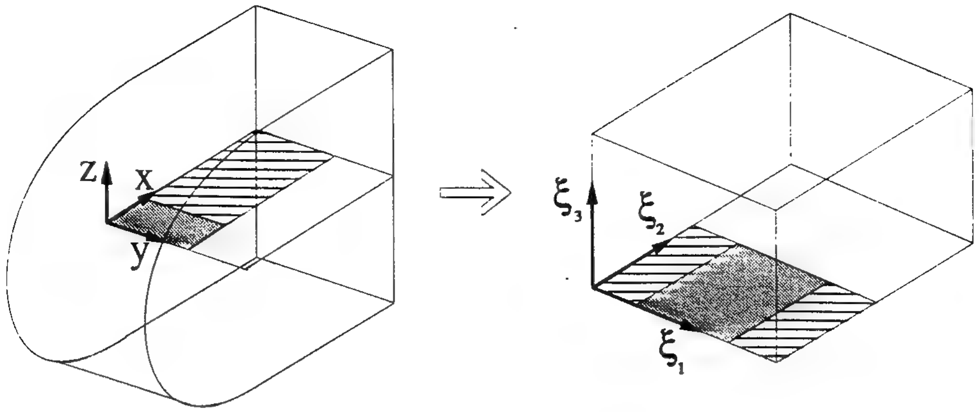

Figure 1. Physical to Computational Coordinate Transformation

Euler Coordinate Transformation

The first step of the grid deformation algorithm is to transform the known grid point deflections to deflections with respect to a body fixed coordinate system. This is an optional step that helps to maintain the orthogonality of grid lines at the physical surface, which is an important property for many CFD codes. If the transformation is used, the deflections on the outer boundary will not be zero upon transformation back to the global coordinate system. A moving outer boundary may be incompatible with physical constraints or may require special boundary conditions. See page 11 for an example of a grid deformed with and without a coordinate transformation.

An Euler transformation is used to transform the grid point deflections to the body-fixed coordinate system. The transformation is defined as three consecutive rotations with respect to the body-fixed coordinate system about a center point $\bar{x}_0$ defined in the global coordinate system. The rotation angles about the x_1, x_2, and x_3 body axes are θ_1, θ_2, and θ_3, respectively, and have corresponding transformation matrices $[\Theta_1]$, $[\Theta_2]$, and $[\Theta_3]$. It should be stressed that the order of rotation is important. The equation for the transformed deflections of points on the physical surface is

$$\{\tilde{u}_b^m\} = [\Theta_1][\Theta_2][\Theta_3]\{\bar{x}_b^m + \bar{u}_b^m - \bar{x}_0\} - \{\bar{x}_b^m - \bar{x}_0\}. \qquad (1)$$

The product of the three transformation matrices is

$$[\Theta_1][\Theta_2][\Theta_3] = \begin{bmatrix} C\theta_2 C\theta_3 & -C\theta_2 S\theta_3 & -S\theta_2 \\ C\theta_1 S\theta_3 - S\theta_1 S\theta_2 C\theta_3 & C\theta_1 C\theta_3 + S\theta_1 S\theta_2 S\theta_3 & -S\theta_1 C\theta_2 \\ S\theta_1 S\theta_3 + C\theta_1 S\theta_2 C\theta_3 & S\theta_1 C\theta_3 - C\theta_1 S\theta_2 S\theta_3 & C\theta_1 C\theta_2 \end{bmatrix}. \qquad (2)$$

where C and S are shorthand for the cosine and sine functions, respectively. If the coordinate transformation is not applied. then Equation 1 simplifies to

$$\left\{ \tilde{u}_b^m \right\} = \left\{ \bar{u}_b^m \right\} \tag{3}$$

since the rotation angles are all zero.

Selection of Marching Directions

The differential equation governing the grid point deflections, described in the next sub-section, requires an initial value. Deflections are computed along grid lines by marching away from the initial value. However, in general, there is not a single computational coordinate that can serve as the marching direction for the entire aerodynamic grid. In such cases, it is necessary to divide the aerodynamic grid into domains within each of which there is a consistent marching direction. Often, the solution of one domain will be used as an initial condition for a second domain.

As an example, consider a two-dimensional C-grid about an airfoil, as shown in Figure 2. The chordwise direction is ξ_1 and the normal direction is ξ_2. Define the wake-cut as the set of points on the $\xi_{2_{mm}}$ grid line that connect the trailing edge with the outer boundary. Deflections on the airfoil surface are known from the CSM code. Deflections are to be determined for the points in the wake-cut, the grid interior, and the outer

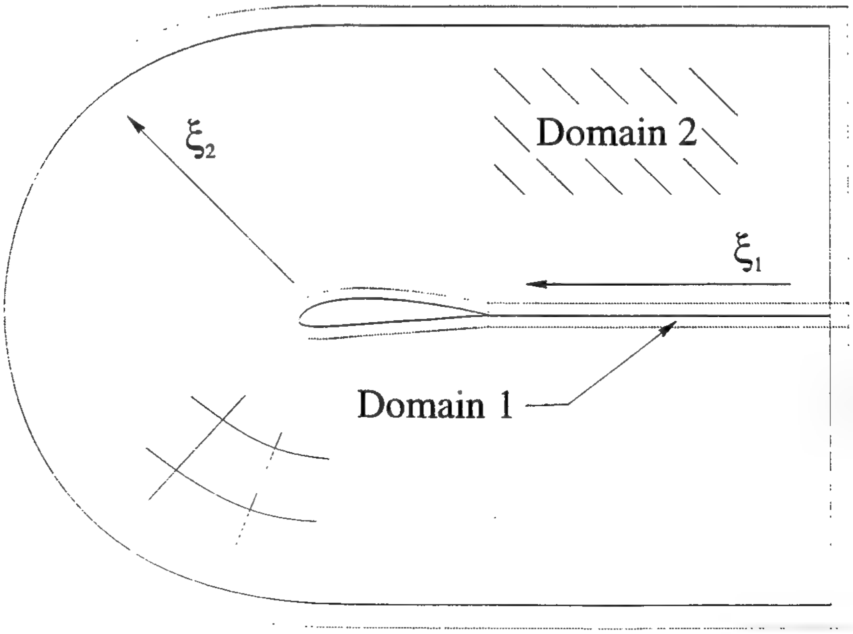

Figure 2. C-Grid About an Airfoil

boundaries. The wake-cut points are chosen to form one domain and the interior points and outer boundary points a second domain. The deflections in the wake-cut domain are obtained by using the deflection of the trailing edge as an initial condition and ξ_1 as the marching direction. The second domain uses both the airfoil and wake-cut grid point deflections as initial conditions and the marching direction is ξ_2.

Solving for Interior Grid Point Deflections

Let ξ_2 be the marching direction. The grid point deflections in terms of the body-fixed coordinate system are determined according to the following ad-hoc non-linear first-order differential equation:

$$\frac{\partial \tilde{u}^m}{\partial \xi_2} = -\nu(\bar{x},\bar{\xi})\left|\frac{\partial \bar{x}}{\partial \xi_2}\right|\frac{\tilde{u}^m}{U}. \tag{4}$$

Each term is described below. The method is considered ad-hoc since the grid deformation problem is not based on any natural phenomenon and is constructed purely based on considerations of computational accuracy and efficiency. Equation 4 is solved by marching along grid lines of constant ξ_1 and ξ_3, with ξ_2 varying from $\xi_{2_{min}}$ to $\xi_{2_{max}}$. The deflection of the $\xi_{2_{min}}$ grid point serves as a boundary or initial condition. The $\xi_{2\,min}$ point will be either a point on the physical surface or a point in another domain for which deflections have already been computed.

Equation 4 produces the desired behavior in the region away from the physical surface: the deflection $\tilde{u}^m$ and the gradient of the deflection $\partial\tilde{u}^m/\partial\xi_2$ are of opposite sign since every term in Equation 4 other than $\tilde{u}^m$ is always positive. Also, as the deflection $\tilde{u}^m$ goes to zero, so does the gradient $\partial\tilde{u}^m/\partial\xi_2$.

Equation 4 can be written as

$$\frac{\partial \tilde{u}^m}{\partial \xi_2} = -\kappa \tilde{u}^m \tag{5}$$

where κ is a function of $\bar{x}$, $\bar{\xi}$, and U:

$$\kappa(\bar{x},\bar{\xi},U) = \frac{\nu(\bar{x},\bar{\xi})}{U}\left|\frac{\partial \bar{x}}{\partial \xi_2}\right|. \tag{6}$$

If κ is replaced by a constant, Equation 5 becomes linear and has an exponentially decaying solution. Thus, it is clear that the rate of decay of the deflections is related to the function $\kappa(\vec{x},\vec{\xi})$.

The spacing function ν is chosen to give the desired grid behavior near the physical surface: surface deflections are propagated some distance into the interior of the aerodynamic grid with little smoothing. The function ν is plotted in Figure 3 and is defined as

$$\nu(\vec{x},\vec{\xi}) = \begin{cases} 0 & r < r_0 - r_1 \\ \frac{1}{2}\left(\frac{r_1 - r_0 + r}{r_1}\right)^3 & r_0 - r_1 \le r < r_0 \\ \frac{1}{2}\left(2 - \left(\frac{r_1 + r_0 - r}{r_1}\right)^3\right) & r_0 \le r < r_0 + r_1 \\ 1 & r_0 + r_1 \le r \end{cases}, \tag{7}$$

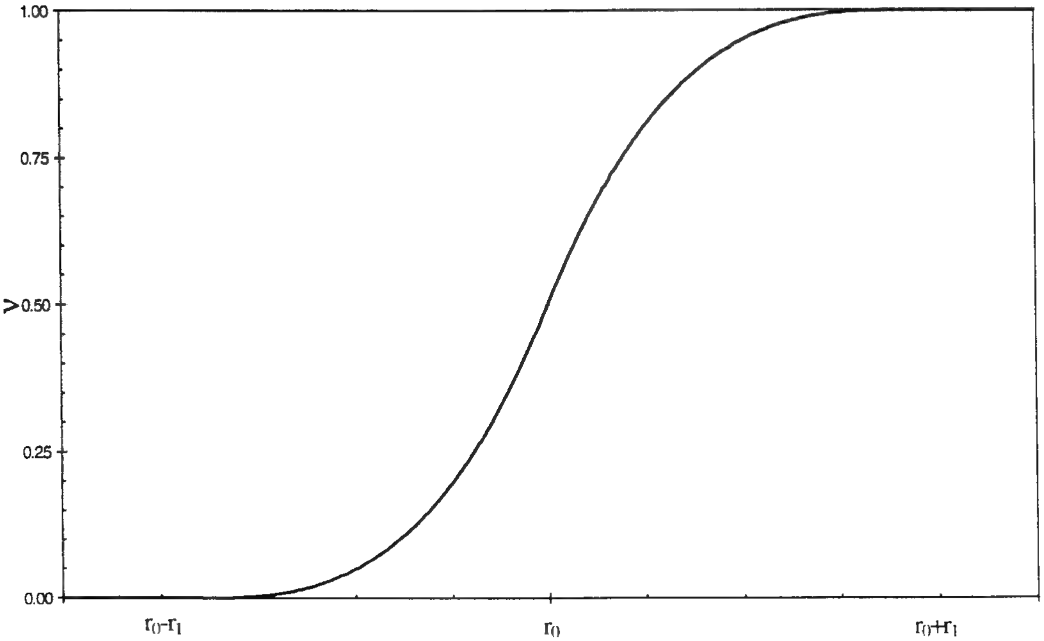

Figure 3. Grid Spacing Function

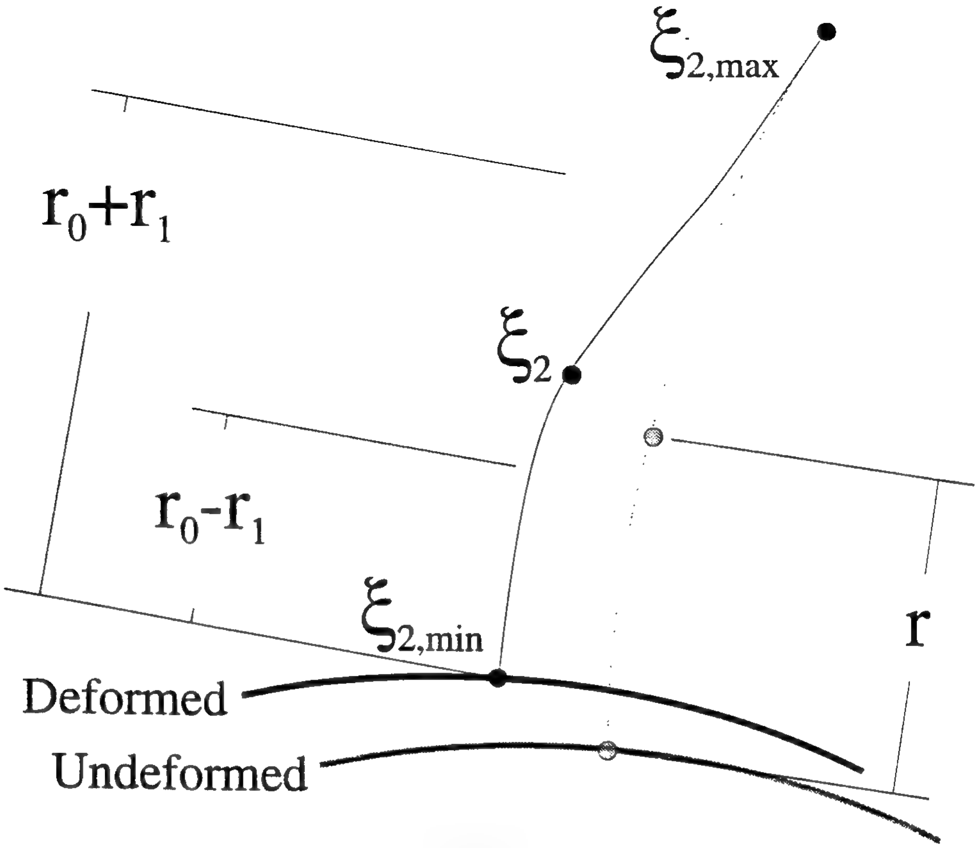

Figure 4. Smoothing Parameters

where r_0 and r_1 are two constants chosen by the user. The constant r_0 is a measure of the distance from the body at which smoothing becomes significant. The constant r_1 is a measure of how rapidly the smoothing becomes significant. No smoothing occurs at distances less than $r_0 - r_1$ away from the surface. The function r is defined as the distance between the point at the base of the grid line and the interior grid point,

$$r = \left| \bar{x}_{\xi_2} - \bar{x}_{\xi_{2\min}} \right|, \tag{8}$$

and is as shown in Figure 4.

The term U is taken to be slightly greater than the magnitude of the maximum deflection on the physical surface. The magnitude $|\partial\bar{x}/\partial\xi_2|$ is evaluated according to the equation

$$\left|\frac{\partial\bar{x}}{\partial\xi_2}\right| = \sqrt{\frac{\partial x_1}{\partial\xi_2}^2 + \frac{\partial x_2}{\partial\xi_2}^2 + \frac{\partial x_3}{\partial\xi_2}^2} \,. \tag{9}$$

The $|\partial\bar{x}/\partial\xi_2|$ and U terms exist in Equation 4 to help prevent the creation of negative cell volumes in the deformed aerodynamic grid.

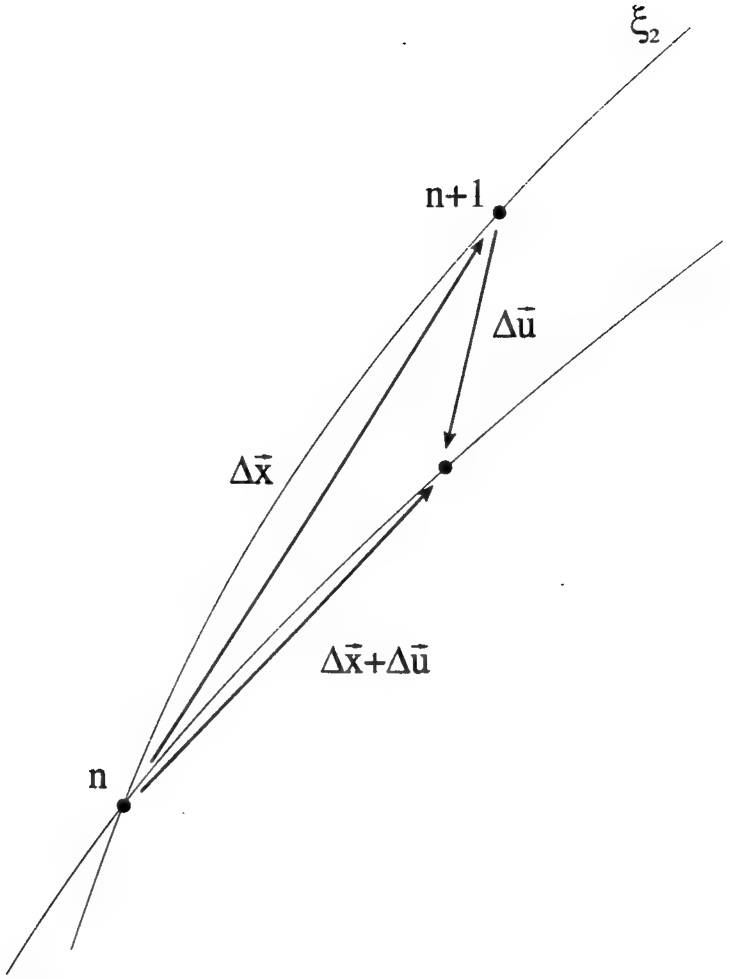

Figure 5. Distances Between Consecutive Grid Points, Undeformed and Deformed

Along a gridline, two consecutive points will not cross if the following condition is satisfied:

$$\left|\frac{\partial \tilde{u}^m}{\partial \xi_2}\right| < \left|\frac{\partial \bar{x}}{\partial \xi_2}\right|. \qquad (10)$$

Consider a ξ_2 grid line, as shown in Figure 5. The deflection at point n is taken to be known and the deflection at point $n+1$ is to be computed. The undeformed distance between the two points is $|\Delta\bar{x}|$. The deformed distance is $|\Delta\bar{x}+\Delta\bar{u}|$. The magnitude of $\Delta\bar{u}$ must be less than the magnitude of $\Delta\bar{x}$ for the ξ_2 grid line to remain uncrossed. Letting the distance between the two points, $|\Delta\bar{x}|$, go to zero, the condition becomes that of Equation 10. Equation 10 is satisfied by Equation 4 since $|\tilde{u}^m/U|$ and ν are always less than one. This property does not prevent two ξ_2 grid lines from crossing. However, grid lines should not cross if the physical surface deforms smoothly and if deflections do not change rapidly over the surface.

Inverse Coordinate Transformation

With all of the grid point deflections known in terms of the body-fixed coordinate system, the grid point deflections in terms of the global coordinate system can be determined by applying an inverse coordinate transformation:

$$\left\{\vec{u}_a^m\right\} = \left[\Theta_3\right]^{\mathrm{T}}\left[\Theta_2\right]^{\mathrm{T}}\left[\Theta_1\right]^{\mathrm{T}}\left\{\vec{x}_a^m + \tilde{u}_a^m - \vec{x}_0\right\} - \left\{\vec{x}_a^m - \vec{x}_0\right\}. \tag{11}$$

Updating Grid Point Positions

With the grid point deflections known for the entire grid, the grid point positions can be updated according to

$$\left\{\vec{x}_{a,b}^{m+1}\right\} = \left\{\vec{x}_{a,b}^{m}\right\} + \left\{\vec{u}_{a,b}^{m}\right\}. \tag{12}$$

Deflections for the $(m+1)^{\mathrm{st}}$ iteration of the CFD-CSM loop are referenced to the grid at the m^{th} iteration.

Numerical Implementation of the Smoothing Equation

The differential equation governing the grid point deflections, Equation 4, must be solved numerically. The partial derivative can be replaced by a first-order forward finite difference. Equation 4 then becomes

$$\tilde{u}_{\xi_2+\Delta\xi_2}^{m} = \left(1 - \Delta\xi_2 \frac{\nu}{U}\left|\frac{\partial\vec{x}}{\partial\xi_2}\right|\right)\tilde{u}_{\xi_2}^{m}. \tag{13}$$

Since the grid point deflections need not satisfy any physical laws, a first-order finite differencing scheme is adequate.

Sample Deformed Grids

This section gives results from two test grids deformed with the method presented in this paper. The first is a two-dimensional O-grid about a cylinder. The second is a three-dimensional C-H grid about a swept wing.

An O-Grid About a Cylinder

Figure 6 shows an O-grid about a cylinder of unit radius. The outer boundary is a distance of 5 radii away from the surface of the cylinder. Grid points are clustered near the surface of the cylinder. In all of the deformed grids about the cylinder, values of $r_0 = 2.5$ and $r_1 = 3.0$ were used in Equation 4.

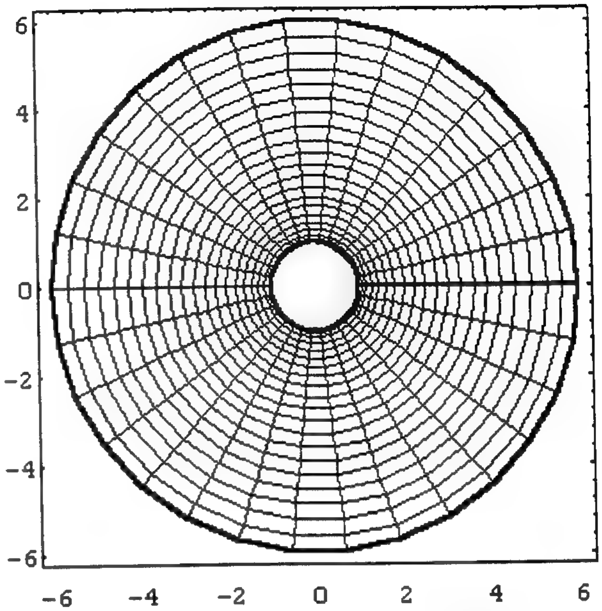

Figure 6. O-Grid About a Cylinder

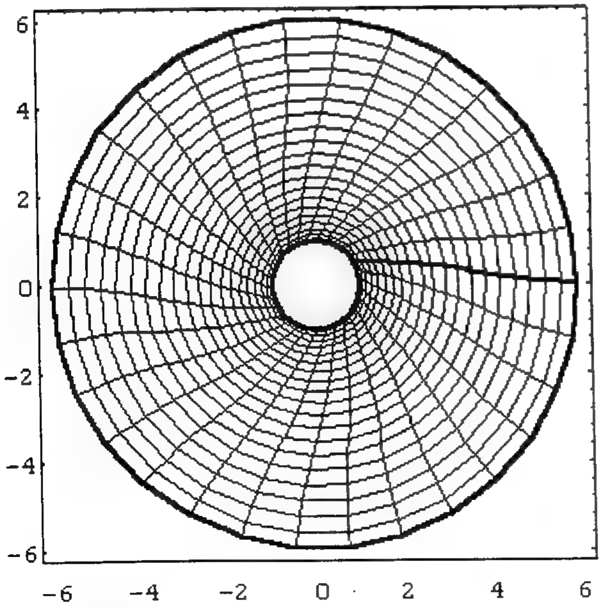

Figure 7. Deformed O-Grid About a Rotated Cylinder without Coordinate Transformation

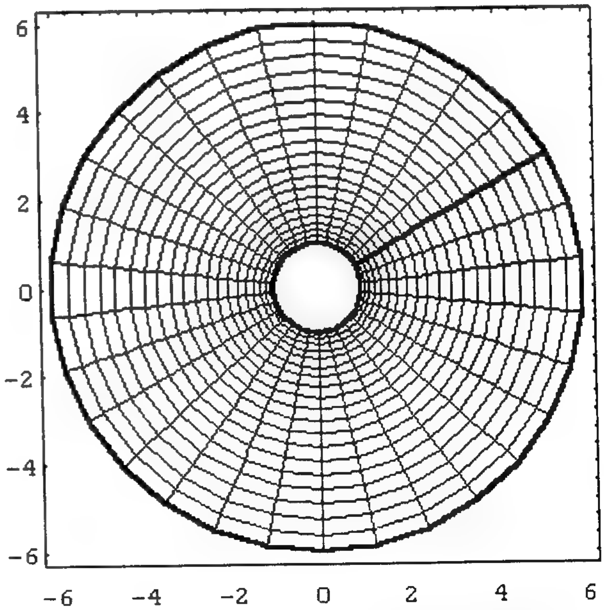

Figure 8. Deformed O-Grid About a Rotated Cylinder with Coordinate Transformation

A deformed grid resulting from a 30° rotation of the cylinder is shown in Figure 7. This grid was obtained without applying a transformation to a coordinate system fixed to the cylinder. The clustering of grid points near the surface of the cylinder has been maintained. However, the grid lines are no longer normal to the surface of the cylinder. Figure 5 shows the effect of the coordinate transformation on the deformed grid. The entire grid is simply rotated by the same amount as the cylinder.

Figure 9 shows the result of a translation of the cylinder. The cylinder is translated half a radius in the negative x-direction and three-halves a radius in the y-direction. In this case, both clustering and orthogonality have been maintained. Since the body-fixed coordinate system is not rotated with respect to the global coordinate system, the transformation would have no effect. The smooth transition between the inner and outer boundaries of the grid is

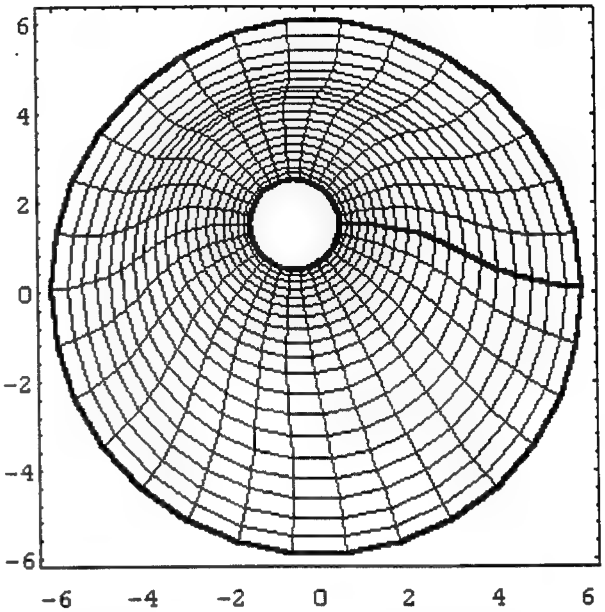

Figure 9. Deformed O-Grid About a Translated Cylinder without Coordinate Transformation

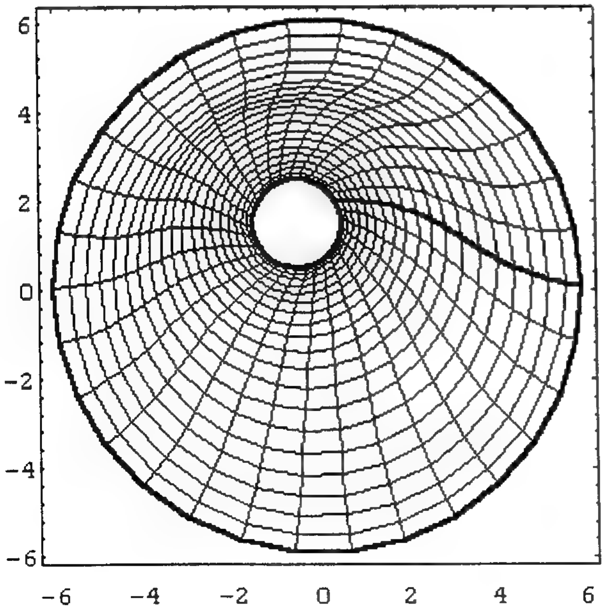

Figure 10. Deformed O-Grid About a Translated and Rotated Cylinder without Coordinate Transformation

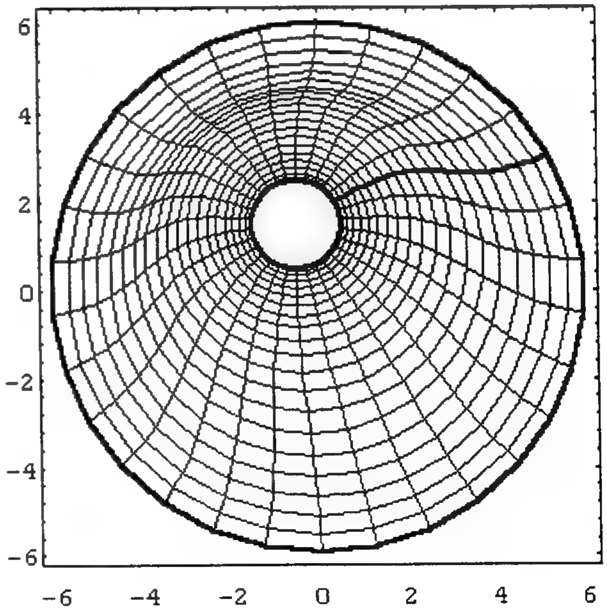

Figure 11. Deformed O-Grid About a Translated and Rotated Cylinder with Coordinate Transformation

easily visible in this figure.

Two final examples of deformed grids about the cylinder are shown in Figure 10 and Figure 11. Here, the cylinder is both translated and rotated. The grid in Figure 10 is produced without applying a coordinate transformation. The grid in Figure 11 was produced using a coordinate transformation. In Figure 10, the outer boundary has not been affected, while in Figure 11 the outer boundary has been rotated by the same amount as the cylinder. Both have maintained grid clustering near the cylinder surface. Orthogonality was only maintained when a coordinate transformation was applied.

These grids show that the differential equation of Equation 4 maintains grid quality under translational motion of the physical surface, but that the quality of the grid deteriorates under rotational motion. The coordinate transformation of Equation 1 improves the quality of the deformed grid in cases with rotational motion, but it does so

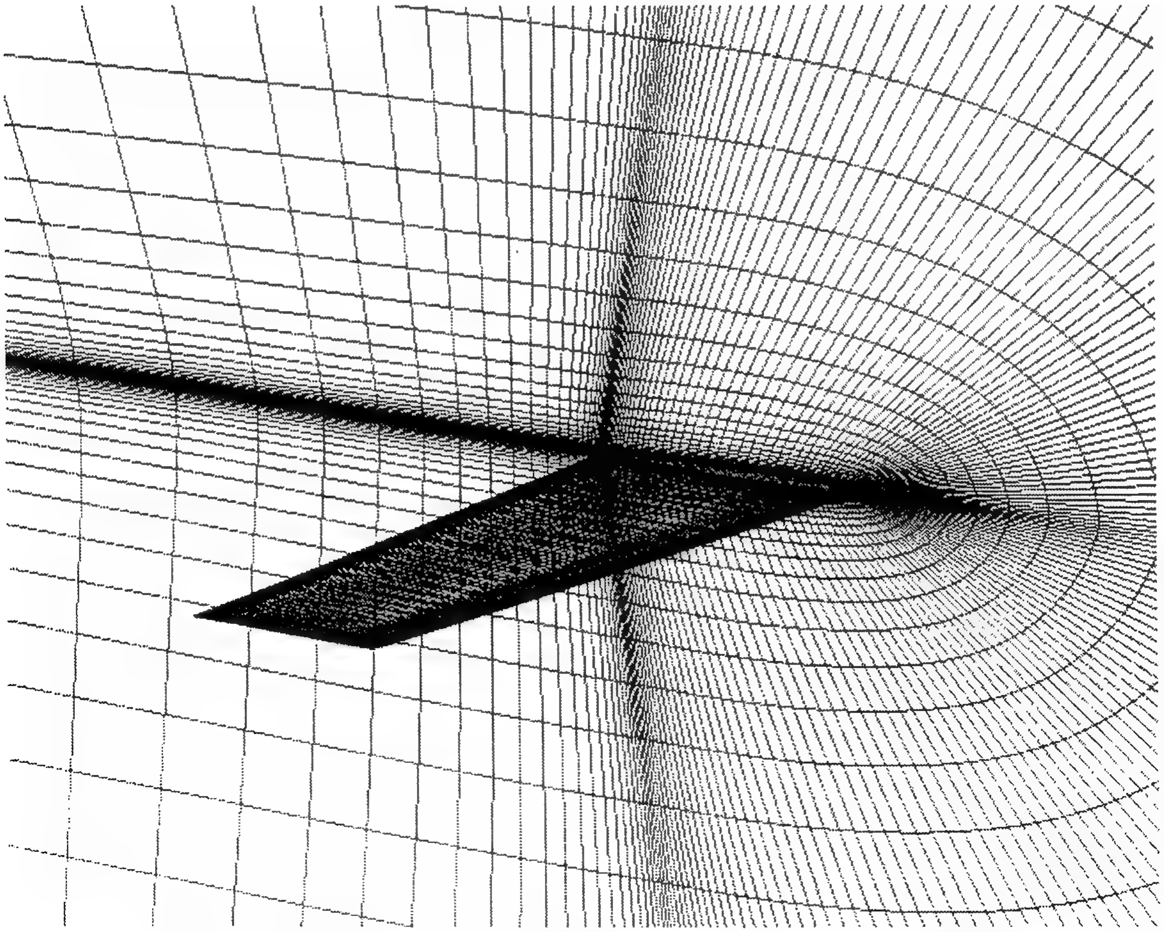

Figure 12. Partial View of a C-H Grid About a Swept Wing

by rotating the entire grid, which moves the outer boundaries.

C-H Grid About a Swept Wing

Figure 12 shows a portion of a C-H grid about a swept wing. The wing was deformed by applying enough of the first bending mode to deform the wing tip by 5% of the root chord. Values of $r_0 = 2.0$ and $r_1 = 0.5$ were used in Equation 4 and no coordinate transformation was applied. For this grid, there was no consistent normal direction in computational space. Therefore, the deformation problem was divided into three domains. First, the deflections of the grid points outboard of the wing tip and in the plane of the wing are computed by marching away from the wing tip in the spanwise, or ξ_2, direction. Second, the deflections in the wake cut are determined by marching in the chordwise, or ξ_1, direction. At this stage, all of the deflections in the $\xi_{3_{\min}}$ plane are known. Finally, the remaining grid point deflections are determined by marching in the ξ_3 direction.

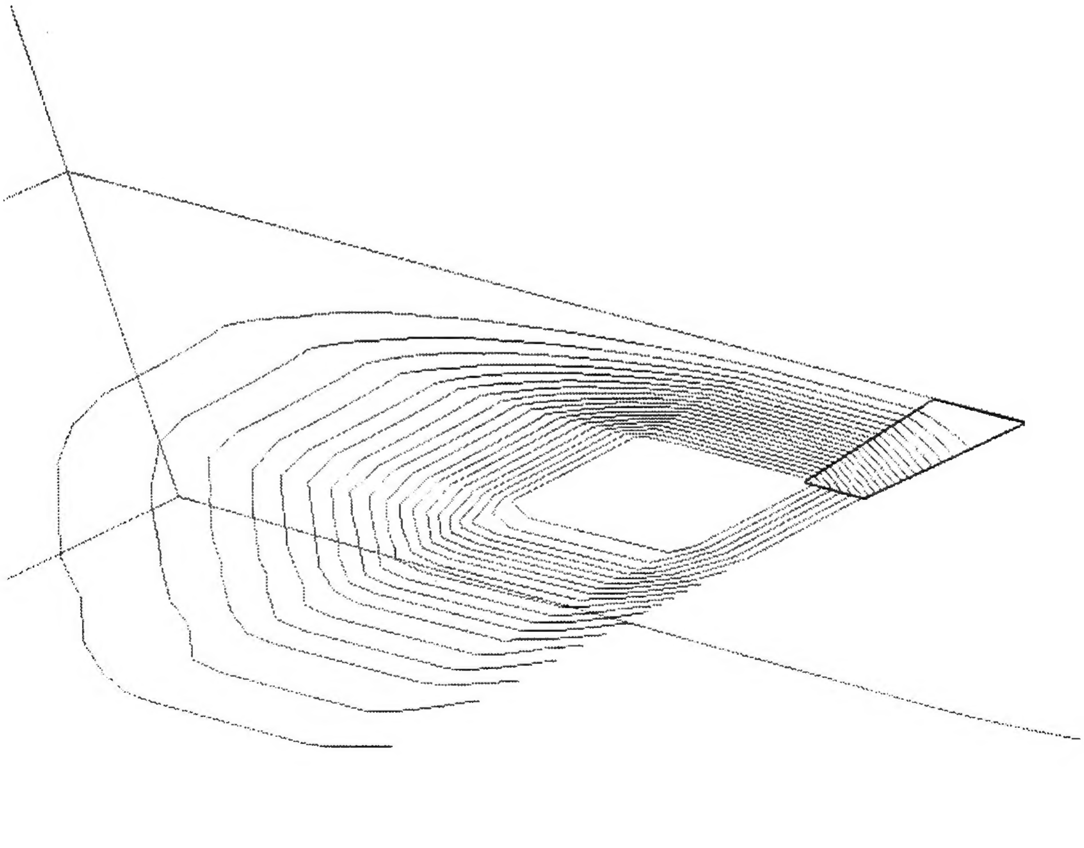

Figure 13. Deflection Contours of a Deformed Grid About a Swept Wing

Figure 13 shows contours of the grid point deflections in the $\xi_{3_{\min}}$ plane of the grid. The wing is outlined with a heavy border. The deflections remain approximately constant for distances less than r_0 away from the wing. The grid deflections then smooth out to zero over a distance of a few root chords. The roughly rectangular shape of the contours is a result of using multiple marching directions.

Conclusions and Recommendations

This memorandum presents an algorithm for deforming an aerodynamic grid given deflections on a physical boundary. The algorithm has been implemented in FORTRAN and has been successfully tested on both two- and three-dimensional grids.

The method is capable of producing reasonable grid deflections in a reasonable amount of computer time for a range of practical problems. The method is faster than both the spring analogy and the global-geometry methods. It is applicable to a variety of physical configurations, including a wing and a wing-body. Also, the method can be applied to a variety of grid schemes, such as C-H grids and H-H grids. It is necessary, however, to code each configuration separately.

The method is limited to reasonably smooth deflections since smoothing is only applied in the direction normal to the surface. A discontinuity in the surface deflections, such as occurs at a deflected flap, is propagated away from the surface without smoothing. The method could be improved by adding a term to the differential equation to smooth such discontinuities in surface deflections.

The method, as presented here, cannot enforce grid point deflections on outer boundaries. In many cases, the grid point deflections will damp to zero, but the algorithm does not impose such a constraint. It should be possible to enforce deflections on multiple boundaries, though, by applying this algorithm iteratively, alternately marching away from the physical surface and the outer boundary. This would make the method applicable to interior flow analyses and to multi-element airfoil analyses.

References

1) Batina, John T., *Unsteady Euler Airfoil Solutions Using Unstructured Dynamic Meshes*, AIAA Paper No. 89-0115, January 1989.

2) Robinson, Brian A., Batina, John T., and Yang, Henry T. Y., *Aeroelastic Analysis of Wings Using the Euler Equations with a Deforming Mesh,* AIAA Paper No. 90-1032, 1990.

3) Snyder, Richard D., *A Global Geometry-Based Grid Deformation Algorithm for Computational Fluid Dynamics for Aeroelasticity*, WL-TM-94-3078-FIBG, 1994.